

软件定义网络控制平面可扩展性研究进展*

张少军, 兰巨龙, 胡宇翔, 江逸茗

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

通讯作者: 张少军, E-mail: zhangsj07@126.com



摘要: 软件定义网络(software-defined networking, 简称 SDN)遵循控制转发分离的设计原则, 其控制平面采用集中的控制逻辑, 在提供灵活、高效的网络控制的同时, 也面临着严重的可扩展性问题。对 SDN 控制平面可扩展性相关工作进行了综述, 首先, 分析了控制平面可扩展性的影响因素并给出改善思路。在此基础上, 从数据平面缓存优化、高性能控制器、分布式控制平面和控制资源优化分配 4 种技术路线出发, 论述了主要的解决方案和研究进展。最后给出总结, 并展望了未来的研究工作。

关键词: 软件定义网络; 控制平面; 可扩展性; 数据平面; OpenFlow

中图法分类号: TP393

中文引用格式: 张少军, 兰巨龙, 胡宇翔, 江逸茗. 软件定义网络控制平面可扩展性研究进展. 软件学报, 2018, 29(1): 160-175. <http://www.jos.org.cn/1000-9825/5316.htm>

英文引用格式: Zhang SJ, Lan JL, Hu YX, Jiang YM. Survey on scalability of control plane in software-defined networking. Ruan Jian Xue Bao/Journal of Software, 2018, 29(1): 160-175 (in Chinese). <http://www.jos.org.cn/1000-9825/5316.htm>

Survey on Scalability of Control Plane in Software-Defined Networking

ZHANG Shao-Jun, LAN Ju-Long, HU Yu-Xiang, JIANG Yi-Ming

(National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002, China)

Abstract: Software-defined networking (SDN) is an emerging paradigm that decouples control and data plane. The control plane advocates a centralized approach of network control, which provides both flexibility and high efficiency, but suffers from serious scalability problems. The state-of-art works on the scalability of control plane in SDN is surveyed in this paper. First, the factors that influence the scalability of control plane are analyzed, and some approaches for improvement are revealed accordingly. Next, current researches and solutions are introduced from the following 4 aspects: Cache optimization on data plane, high-performance controller, distributed control plane and control resource allocation optimization. Finally, a brief summary of current technology is provided along with a discussion on future research directions.

Key words: software-defined networking; control plane; scalability; data plane; OpenFlow

人类社会已经进入网络无所不在的信息时代, 截至 2015 年 12 月, 全球互联网用户突破 33 亿^[1], 全球流量也将于 2018 年达到 1.6×10^{21} 字节^[2]. 互联网能够快速发展, 主要得益于其优良的体系结构和网络技术, 然而, 伴随着推广的深入和网络规模的不断扩大, 爆炸式的增长, 使其自身结构的僵化问题越来越明显, 互联网原有的结构反而成为阻碍它进一步发展的最大障碍^[3,4].

* 基金项目: 国家重点基础研究计划(973)(2013CB329104); 国家自然科学基金(61521003, 61502530, 61372121); 国家高技术研究发展计划(863)(2015AA016102)

Foundation item: National Key Basic Research Program (973) (2013CB329104); National Natural Science Foundation of China (61521003, 61502530, 61372121); National High Technology Research and Development Program of China (863) (2015AA016102)

收稿时间: 2016-06-16; 修改时间: 2017-02-03, 2017-04-07; 采用时间: 2017-06-14; jos 在线出版时间: 2017-07-12

CNKI 网络优先出版: 2017-07-12 16:16:47, <http://kns.cnki.net/kcms/detail/11.2560.TP.20170712.1616.015.html>

斯坦福大学的学者于 2008 年提出 OpenFlow^[5],并在 2009 年将其扩展为 SDN(software-defined networking)概念^[6].SDN 将控制平面与数据平面分离,使得控制平面和数据平面可以独立演进.SDN 还设计了统一的、标准化的南向接口以及逻辑集中的、开放的、可编程的控制平面,使得网络管理人员可以掌握全局网络视图,对网络进行灵活的配置和管理,降低了整体运营成本,并且方便了新协议的开放和部署.SDN 自提出以来,相关研究和产业化应用迅速展开^[7-10],为未来网络的快速演进提供了新的思路.

典型的 SDN 架构从上到下包含应用平面、控制平面、数据平面 3 层,其中,数据平面由交换机等设备组成,提供可编程的快速转发能力;控制平面^[11]是指逻辑集中的控制器,负责维护全网视图并运行控制策略,控制平面通过南向接口(主要是 OpenFlow 协议^[12])对数据平面进行控制和监测,通过北向接口向应用平面提供网络编程能力;应用平面包含路由、流量工程、故障恢复、网络虚拟化等网络应用,用户可以通过北向接口从控制平面获取全局网络视图,根据该信息编写高层网络策略对网络事件进行处理.Google B4^[13]是 SDN 最成功的应用案例,通过在数据中心 WAN 中部署基于 OpenFlow 的集中式流量工程服务,B4 将链路利用率提高到接近 100%.

随着 SDN 应用范围的扩展和网络规模的扩张,控制平面作为 SDN 网络的关键组成部分,其处理能力成为 SDN 网络的性能瓶颈,极大地影响了 SDN 的推广和演进.例如,包含 1 500 台服务器的数据中心集群平均流到达速率为 100K/s^[14],拥有 100 个交换机的网络其流到达速率峰值可达 10M/s^[15],而最初的 NOX 控制器^[16]在保障其 10ms 响应时间的前提下,每秒仅能响应 30k 个流处理请求^[15].即使是当前主流的运营商机控制 ONOS,其吞吐量也尚未达到 1M/s 的设计目标^[17].当前,SDN 控制平面面临着严重的可扩展性问题^[18].

可扩展性是保持互联网体系结构不断发展的原动力之一,可扩展的控制平面对 SDN 网络意义非凡.首先,随着网络流量的不断增加,可扩展的控制平面能够提供足够的处理能力以满足网络的需求,防止发生服务拒绝以影响网络服务质量;其次,随着网络拓扑规模的扩大,可扩展的控制平面仍然能够以足够的处理能力提供优质的服务,使得 SDN 网络的应用场景从局域网、数据中心拓展到广域网乃至整个互联网;最后,可扩展的控制平面能够提供对细粒度流处理的良好支持,从而促进网络协议的创新和网络的不断演进.

本文第 1 节在分析 SDN 基础架构和工作流程的基础上,结合分布式系统可扩展性的定义及衡量标准,给出改善 SDN 控制平面可扩展性的思路.在此基础上,第 2 节~第 5 节对现有的几类改善控制平面可扩展性的技术路线进行详细阐述.第 6 节给出总结,并对未来研究提出展望.

1 SDN 控制平面的可扩展性

1.1 控制平面可扩展性的定义

SDN 网络将控制功能从传统网络中解耦、合并,并将其置于远端的逻辑集中的控制平面,这使得网络管理人员可以根据全局网络视图设计网络应用并执行控制策略.然而,这些原本存在于本地的控制功能由于集中到远端的控制平面,使得控制平面的负载随着网络规模的扩大(主要是流量的增长和节点、链路数量的增加)而提升,控制平面的处理能力成为 SDN 网络的瓶颈,即,SDN 控制平面存在可扩展性问题.

一个系统的可扩展性^[19,20]定义为:用户和系统资源的增加不会导致系统性能的明显下降,也不会增加系统管理的复杂性.据此,SDN 控制平面的可扩展性可进一步定义为:当网络规模和复杂度增加后,该控制平面能够处理由此带来的更多的网络事件请求.同时,其响应时间不会发生明显的下降.

一般地,根据生产率来衡量系统的可扩展性^[19].当系统规模为 k 时,其生产率 $F(k)$ 定义为

$$F(k) = \frac{\lambda(k)}{T(k) \cdot C(k)} \quad (1)$$

其中, $\lambda(k)$ 是指系统的吞吐量, $T(k)$ 是指平均响应时间, $C(k)$ 是指系统成本.

由此,可进一步定义网络规模从 k_1 增长到 k_2 的情况下,系统可扩展性的评价指标 $\psi(k_1, k_2)$ 为

$$\psi(k_1, k_2) = \frac{F(k_2)}{F(k_1)} = \frac{\lambda(k_2)}{\lambda(k_1)} \cdot \frac{T(k_1)}{T(k_2)} \cdot \frac{C(k_1)}{C(k_2)} \quad (2)$$

当该指标大于某一阈值时,可认为系统具有良好的可扩展性.从公式(2)可以看出:系统的可扩展性与吞吐量

变化成正比,与响应时间变化和成本投入变化成反比.

这一评价指标也可用于衡量 SDN 控制平面的可扩展性^[21].在 SDN 网络中,控制平面的吞吐量是指控制平面处理网络请求的能力,也称为控制平面的容量,由其计算、存储、带宽资源支持;响应时间具体指的是控制平面对网络请求的响应时间;控制平面成本则包括控制平面的部署和运行成本等.

1.2 控制平面的负载

SDN 网络控制平面的负载主要来源于对网络请求的处理,典型的网络请求包括数据转发、网络视图管理、网络故障处理等.一般来说,处理转发请求是满足网络传输需求的最基本也是最主要的功能,对转发请求事件的处理也是控制器最主要的负载^[15],因此,我们采用转发请求事件的到达速率来衡量控制平面的负载.

OpenFlow^[12]是一种典型的 SDN 架构.在 OpenFlow 协议中,转发请求事件主要指的是 PACKET_IN 事件.如图 1 所示,SDN 网络对转发请求事件的处理流程包括 5 步.

- (1) 新流到达:当一个数据流的首 1 个数据包到达交换机时,若交换机中没有匹配的流表项,则生成转发请求事件;
- (2) 上传转发请求:将转发请求上传到控制平面;
- (3) 计算转发规则:控制平面根据全局网络视图和高层网络策略计算转发规则并生成流表项;
- (4) 下发流表项:控制平面将流表项下发给转发路径涉及的交换机;
- (5) 安装流表项:收到流表项的交换机安装流表项.

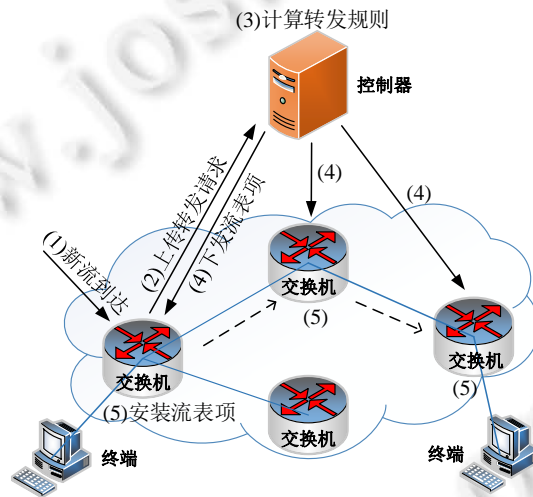


Fig.1 Flow initiation in OpenFlow based SDN

图 1 基于 OpenFlow 的 SDN 网络中转发请求事件的处理流程

可以看出,控制平面对转发请求事件的响应时间主要包括两个部分.

- (1) 转发请求和流表项(统称控制流量)在交换机和控制平面之间的传输时间,该时间主要与交换机到控制平面的距离有关;
- (2) 控制平面的服务时间,根据排队理论,该时间与控制平面容量及当前负载有关.

1.3 改善控制平面可扩展性的思路

根据第 1.1 节中所述控制平面可扩展性的评价指标及其与控制平面容量、响应时间、成本之间的关系,可以得到以下 4 种改善控制平面可扩展性的思路.

- (1) 优化管理交换机中缓存的流表项,以减少数据平面发起的网络请求,进而降低控制平面的负载.该思路降低了单位网络规模产生的控制负载,实际上提高了控制平面的可扩展性.同时,该方式由于控制

平面的负载减少,使得响应时间缩短;

- (2) 提高控制平面的处理能力,即提高吞吐量;
- (3) 减小控制平面和交换机之间的传输时延,即缩短响应时间;
- (4) 提高现有控制资源的利用率,使单位控制资源投入带来的可扩展性改善程度提高。

SDN 一经问世,学术界和产业界就对其控制平面的可扩展性展开了一系列的研究实践.在接下来的 4 节中,我们将基于本节所总结的 4 种思路对当前改善控制平面可扩展性的技术进行论述。

2 数据平面缓存优化

控制平面将流表项下发到交换机后,该流表项会缓存在交换机中,如果新到达的数据包与交换机中的流表项匹配,则不需要再向控制平面提交转发请求.这种通过缓存减轻控制平面负载的思路能够极大地改善控制平面的可扩展性,在传统网络设备如路由器中就已得到广泛的应用.在 SDN 网络中,我们以 OpenFlow 交换机为例,相关机制提供了对流表项缓存的支持.一种最常用的思路是提前主动地在交换机中安装流表项,但这无疑将使大量流表资源无法释放,增加了交换机的负担,带来数据平面的可扩展性问题.因此,有必要结合 SDN 的可编程特性对数据平面的缓存优化进行深入的研究.本节对现有的两类能有效改善控制平面可扩展性的数据平面缓存优化技术展开综述。

2.1 流表项存储优化

SDN 交换机当前普遍采用三态内容寻址存储器(ternary content addressable memory,简称 TCAM)进行流表项存储^[22],TCAM 的高成本和高能耗使得 SDN 交换机的流表项存储空间通常较小.当网络中流较多时,对应的流表项较多,很容易耗尽交换机的流表资源,使得后续流无法在交换机中安装对应的流表项,造成大量的数据包因交换机无规则匹配而直接发送到控制平面,给控制平面带来很大的处理压力.因此,有必要对流表项的存储机制进行优化设计,提高数据平面流表资源的利用率,进而改善控制平面的可扩展性.常用的流表项存储优化方法包括规则压缩和超时释放。

(1) 规则压缩

规则压缩^[23,24]是一种在保留原有规则语义的前提下减少所需规则数目的技术.在 OpenFlow 中,规则以流表项的形式存在,包含匹配域(match)和动作(action)两部分.匹配域标识数据流,而动作规定对该流的处理方法.当多条流表项的动作字段相同时,可以通过在匹配域使用通配符规则计算出匹配域合并方案,从而将这些流表项压缩.简单的示例如图 2 所示.两条流表项的动作相同,而源 IP 地址可以使用通配符合并,且正好包含原来两条流表项的语义.基于通配符可以扩大流表项的语义表达范围,适配更多的数据流,从而减少对数据平面流表资源的占用。

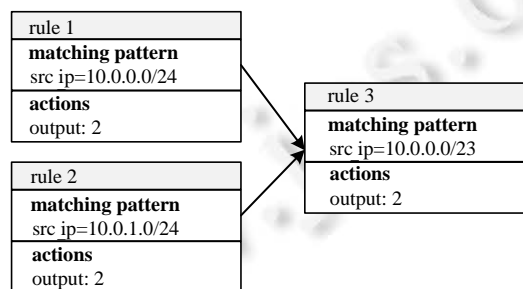


Fig.2 An example of rule compression

图 2 规则压缩示例

传统的路由器使用前缀匹配算法对路由表项进行压缩,但是,由于 OpenFlow 流表项中除了包含目的地址外,还支持源地址、端口、VLAN 等多个匹配域,因此这一方法直接应用于 OpenFlow 规则压缩时效果有限.Bit

Weaving 方法^[25]通过比特置换和比特合并可以实现非前缀规则压缩,同时提供了非常优越的压缩效果,可用于 OpenFlow 规则压缩.文献[23]使用 Espresso 启发式算法对通配符表达式进行逻辑最小化处理,从而实现非前缀匹配的规则压缩.文献[26]则基于 C4.5 决策树算法提供了一种在线的 OpenFlow 规则压缩技术.

规则压缩技术的基础是控制平面不需要对所有数据流进行严格的细粒度处理,而是仅仅根据用户需求选择特定的数据流进行细粒度处理,其他数据流则可以采取相同的方式处理.当压缩规则与细粒度处理规则同时匹配时,可以采用规则优先级进行区分,优先执行细粒度处理规则.

规则压缩虽然可以减少控制平面的负载,但却存在以下问题.首先,规则压缩屏蔽了数据流的细节,使得控制平面难以实现细粒度的流监测;第二,最优化通配符规则压缩是 NP 难的^[27],较高的计算复杂度会增加控制器的响应时间,在设计规则压缩算法时,应对其计算复杂度和压缩效果进行权衡考虑.此外,规则压缩容易使相应流表项对应的业务量过大,在该交换机及相应链路上造成拥塞.文献[28]设计了一种动态路由机制,该机制综合考虑了数据平面链路带宽和交换机流表项存储空间,通过动态调整转发规则的放置,避免造成交换机和链路的超载,同时最大化网络的吞吐量.这种机制利用数据平面空闲的交换机和链路来承担热点交换机和链路的负载,可用于缓解规则压缩的缺陷.

(2) 超时释放

由于数据平面的流表资源受限,OpenFlow 中设计了超时机制^[29]来主动删除缓存的流表项,为新流的到达预留流表项存储空间.超时机制包含硬超时(hard_timeout)和空闲超时(idle_timeout)两种.其中,硬超时规定了流表项在交换机中的最大生存时间,而空闲超时则表示没有数据流匹配后流表项的生存时间.如图 3 所示:从数据平面的可扩展性出发,超时阈值应设置得尽量小,以减少旧规则对流表资源的占用;但是过小的超时阈值会使历时较长的流被截断为数段,每段流都触发对控制平面的转发请求,给控制平面带来额外的负担.

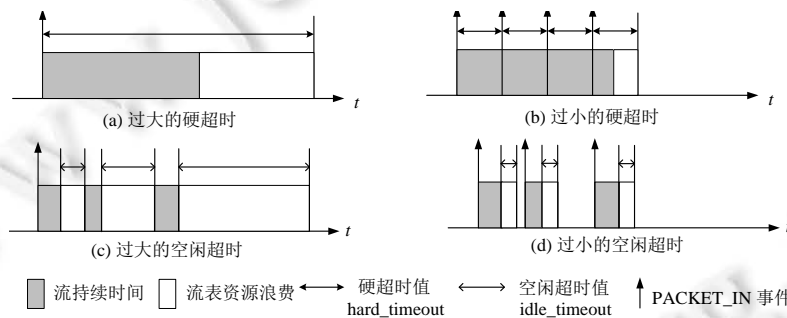


Fig.3 Effect of small and large timeout

图 3 过大和过小的超时值的影响

由于不同数据流的持续时间不同^[30],设计动态变化的超时阈值显得更为合理.对持续时间短的流,设计小的超时阈值,可以及时删除已不再有效的规则,节约流表资源,改善数据平面的可扩展性;对持续时间较长的流,设计与流持续时间相当的大超时阈值,可以防止由于规则超时删除导致同一流的重复请求,降低控制平面的负担,改善控制平面的可扩展性.

AHTM^[31]和 TimeoutX^[32]分别基于排队系统建模和实际测量研究了数据流的持续时间特征,并结合数据平面流表资源和控制平面处理资源的消耗,对 OpenFlow 的硬超时机制进行了优化设计.文献[33]则基于 ON/OFF 流量模型对 OpenFlow 的空闲超时机制进行分析,并给出了符合资源限制的超时阈值的上下界.文献[34]综合考虑了控制器负载和交换机流表资源两项成本,对新到来的流是否缓存、设置多大的超时阈值等策略进行折中处理和优化设计.

2.2 控制功能授权

软件定义网络基于控制数据分离的原则,数据平面为了保障独立演化和通用性,往往只实现最基础的匹配

转发功能。

控制平面可扩展性问题的主要原因是:原本位于本地的控制功能被解耦合并至远端的控制平面,而数据平面规模扩大使得控制平面的处理能力不足或距离过远.因此,最直观的做法是将部分控制功能授权给本地数据平面,使部分网络请求在本地即可得到处理,同时减轻控制平面负载.下面对几种控制功能授权技术进行介绍.

DIFANE^[35]中,控制器将数据平面分域,在每个域中选出权威交换机,并根据网络策略和全局网络视图在权威交换机上主动安装权威规则.普通交换机由权威交换机控制,向权威交换机提交转发请求.权威交换机根据自身安装的权威规则向普通交换机安装转发规则.DIFANE 将控制平面的负载分担给了权威交换机,但是该方法需要权威交换机具备规则的生成与安装功能,传统的 OpenFlow 交换机并不支持这一功能,无法得到广泛推广.同时,与规则压缩方式类似,DIFANE 也屏蔽了控制平面对数据流的细粒度监测,削弱了控制决策的精度.

DevoFlow^[36]的思想与规则压缩类似,该技术认为:完全的细粒度控制并不是网络的目标,大部分数据流不需要细粒度的控制,对它们的处理应该放在数据平面;控制器只需保持对部分关键流的处理和监测即可. DevoFlow 采用了规则复制方式来实现对数据流的细粒度监测,它在交换机中安装的是带有通配符的流表项,流表项中的 action 字段中包含 CLONE 标识.若该标识清零,则所有符合该流表项的数据流可看作是同一个数据流而进行监测;否则,该流表项按不同的数据流复制,从而细化各个匹配流的监测信息. DevoFlow 同样需要修改 OpenFlow 交换机的硬件结构,不具有通用性.

FAST^[37]考虑到许多网络应用可表示为状态机,整个应用涉及到多种数据流的参与(如 TCP 协议流程).如果每个数据流均向控制器提交,会增加控制平面的负担和网络应用的响应时间.针对这一问题,FAST 设计了一种新的交换机结构,使得控制平面能够向交换机安装对应网络应用处理流程的状态机,使其自动地对网络应用生成一系列的处理规则. SDPA^[38,39]进一步完善了数据平面的状态机处理思路,并给出了基于“match-state-action”的数据平面框架及完整的软硬件实现方案.这些方案均需对 SDN 数据平面乃至控制平面架构进行新的抽象,也是数据平面下一步的演进方向.

与上述需要改变当前数据平面架构以实现必要的控制功能的技术不同,文献[40]从理论上证明现有的 OpenFlow 规则可通过组合来实现大部分确定型图灵机可实现的网络应用,由此,在当前 SDN 结构条件下控制器可以将更多的控制功能授权给交换机,使一些时延敏感的网络任务在数据平面即可得到处理.

实际上,当前业界在应用 SDN 技术时也越来越多地采用控制功能授权这一思路.例如,在部署 SDN 网络时,在某些交换机的通用处理器上安装轻量级控制器,实现一些仅需本地视图信息的控制功能.常用的软件交换机 Open vSwitch^[41]即包含了一个 OpenFlow 控制器的参考实现.

3 高性能控制器

仅仅通过缓存优化来减轻控制平面的负载并不能从根本上解决 SDN 控制平面的可扩展性问题,其原因在于:单一结构集中的控制平面处理能力不能满足日益增长的网络规模和网络需求,遇到了性能瓶颈.最初的 NOX 控制器^[16]在保障其 10ms 响应时间的前提下,每秒仅能响应 30k 个流请求^[15].因此,设计高性能控制器、提高控制平面的处理能力,是一种更为直接的选择.这种改善控制平面可扩展性的方式是对控制平面性能的垂直扩展,也称为 scale-up 方式.

Maestro^[42]对 NOX 的编程模型进行了分析,指出,其单线程的编程模型无法充分发挥多核处理器的并行计算性能.此外,NOX 在发出消息时,各个消息调用独立的套接字传输,这一传输时间占了整个网络请求处理时延的 80%.针对这些问题,Maestro 在设计控制器时添加了多线程抽象管理结构,以充分发挥多核处理器的并行计算性能,提高控制器的处理能力;并设计了消息批处理机制,使所有消息按目的地分组进行批处理,提高了控制器的 I/O 性能,降低了消息传输时延.基于多线程处理、高性能 I/O 技术, Maestro、Beacon^[43]以及 NOX 的多线程版本 NOX-MT^[15]均在吞吐量及响应时间上有了明显的提升.

多线程编程模型的瓶颈在于线程管理的复杂度,该复杂度会随着网络规模的扩大和控制负载的增加急剧提升,使处理性能达到瓶颈. McNettle^[44]将多核控制器看作一组逻辑集中的控制器簇,每个核作为一个控制器,交

交换机由固定的核管理,核之间通过共享内存维持一致的全局网络视图,并结合 Haskell 的并发管理工具实现并发调度.如此设计,使得 McNettle 更适合安装在高性能多核服务器中,在较大的数据中心网络保持良好的响应性能(配置 46 核的服务器可管理 5 000 个交换机,支持的峰值流到达速率为每秒 14M).

对于众多中等规模的网络来说,一般使用一个控制器即可完成相应的控制功能.然而,对于大规模网络,仅依靠多线程等处理方式将无法保证性能和管理灵活性.一方面,复杂的并行和并发管理无力胜任更大的网络规模和控制负载,存在性能瓶颈;另一方面,较大规模的网络一般横跨广阔的地理区域,仅部署一个控制器会对距离较远的交换节点产生较高的响应延迟;最后,由于用户行为的随机性,控制负载具有随机突发特性^[45],高性能控制器技术对控制资源缺乏灵活、高效的管理,在应对控制负载的实时随机变化时不具备经济性和可操作性.这些都限制了高性能控制器技术的可扩展性改善空间.

4 分布式控制平面

提高控制平面的处理能力不仅可以通过提高单个控制器的性能,也可以通过增加控制器的数目来实现.将数据平面分域,每个控制器控制一部分网络视图,将整体的网络负载分发给多个控制器进行处理,也能起到提高控制平面整体处理性能、改善控制平面可扩展性的作用.此外,通过对多个控制器进行优化放置,对交换机与控制器的映射关系进行优化配置,可以拉近控制平面与数据平面的距离,降低控制信令的传输延迟,改善控制平面的可扩展性.而且,当控制平面需要扩充控制资源以应对网络规模的扩张时,分布式控制平面可以通过增加额外的控制器实体来实现,这比在单个高性能控制器上扩充控制资源成本低廉,且灵活便捷.不同于提高控制器性能这种垂直扩展方式,由于分布式控制平面是对控制平面资源的水平扩展,也称为 scale-out 方式.

按照组织形式的不同,分布式控制平面可以分为逻辑集中、物理分布的控制平面,完全分布式控制平面,以及分层控制平面.下面分别进行介绍.

4.1 逻辑集中、物理分布的控制平面

在对单一结构集中控制的控制平面进行分布式扩展时,为了维持控制器基于全局网络视图进行集中控制的优势,分布式控制平面往往采取逻辑集中、物理分布的控制平面,在位于不同位置的控制器之间维持全局一致的网络视图.这种控制平面的结构如图 4 所示,其最典型的设计包括 Onix^[46]、HyperFlow^[47]、DISCO^[48]、ONOS^[17]、OpenContrail^[49]、SCL^[50]等.其中,ONOS 和 OpenContrail 在产业界已经有了广泛的应用.

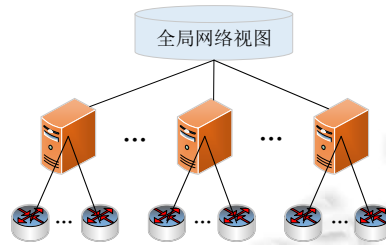


Fig.4 Logically centralized but physically distributed control plane

图 4 逻辑集中物理分布的控制平面

逻辑集中、物理分布的控制平面面临的主要挑战是如何维持一致性的网络视图^[51,52]、保障网络视图的及时更新.文献[53]以 SDN 网络链路负载均衡应用为例,评估了不一致的网络视图对负载均衡性能的影响.该文结论指出,不一致的网络视图会显著降低集中式网络应用的处理性能.文献[52]对分布式控制平面中不同的一致性约束对 SDN 性能和可用性的影响进行了量化分析,为一致性等级的配置提供了参考.一致性问题也是现有的分布式系统迫切需要解决的问题.

Onix 面对这一问题时,针对不同的网络需求采用了两种网络视图更新分发方案.对网络事件更新缓慢、一致性要求较高的情况,采用事务处理数据库模式;对高度动态的网络,则采用 DHT 方式构建快速响应的分布式

结构.HyperFlow 则采用分布式文件系统 WheelFS 同步全局网络视图,并通过发布-订阅模式传播网络事件.这种机制使得 HyperFlow 仅适用于网络事件更新缓慢、一致性要求不高的网络.ONOS 在演进至第 2 个原型后,采用低延迟的 RAMCloud 分布式云数据库存储全局网络视图,并基于 Hazelcast 设计了发布订阅机制实现网络事件分发,经过一系列改进后,极大地降低了控制平面的处理延迟,推动了 SDN 在大型运营商网络的应用.

4.2 完全分布式的控制平面

逻辑集中、物理分布的控制平面为了维护全局网络视图,必须不断地执行更新操作,当网络高度动态时,这种更新操作会带来沉重的网络负担.针对这一问题,一种解决思路是各分布式控制器仅维持本地网络视图,并以分布式的方式执行网络控制.这与传统互联网的控制方式类似,但是由于 SDN 控制转发分离的特性,具有更大的灵活性和易于管理的优势.

完全分布式的控制平面结构如图 5 所示.每个控制器仅仅管理自己域内的网络事件,并根据本地网络视图做出相应的决策,不再试图同步全局网络视图,而是只通过与邻域控制器协作交互完成全局请求处理^[54,55].因此在执行全局路由优化等网络应用时,虽然网络中每个请求都会有控制器处理,但是由于缺乏全局网络视图,跨域的网络请求无法得到全局最优的处理,只能通过设计专门的分布式优化算法和邻域控制器交互协议实现较好的处理效果^[55],而且收敛速度比逻辑集中的控制平面要慢.另一方面,这种控制平面结构很适用于不同利益实体的网络之间的通信,实体之间能够自主地决定可以提供给对方的信息,并隐藏自己的网络细节.这类控制平面的典型代表包括 Devolved Controller^[54]和 WE-Bridge^[56],SDNi^[57]则提供了一种标准化的控制器间信息交互协议.

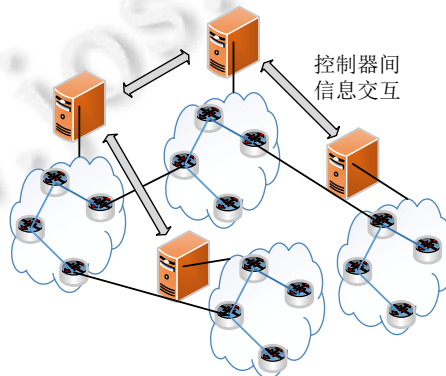


Fig.5 Completely distributed control plane

图 5 完全分布式的控制平面

4.3 分层控制平面

前两种控制平面中,每个控制器的角色及功能完全相同.而分层控制平面中,每个控制器的功能随角色而不同.典型的分层控制平面架构如图 6 所示.

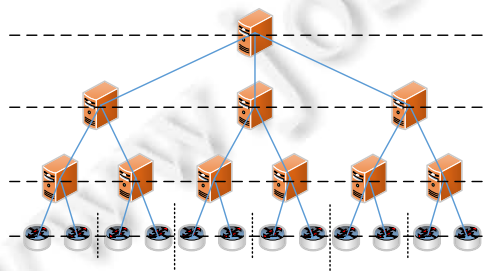


Fig.6 Hierarchical control plane

图 6 分层控制平面

分层控制平面中,数据平面分域管理,每个域配备一个本地控制器,本地控制器仅维持本地网络视图,处理本地的网络事件.进一步地,将本地控制器及其控制域聚合抽象为超级节点,对超级节点形成的平面也进行分域管理,每个域配置一个控制器,每个控制器维持本域超级节点构成的抽象网络视图,并处理本域内的网络事件.以此类推,直至一个顶层控制器.最终,分层控制平面组织为树形控制结构,最底层的交换机为叶子节点,往上的父节点均为控制器,直至作为根节点的顶层控制器.

分层控制平面中,网络控制以递归的形式进行,若网络请求在本域即可解决,则完全由本域控制器处理;否则,逐级上传,直到可以涵盖该网络请求所涉及的拓扑,再逐级计算并下发流表项.层与层之间仍采用标准南向接口,如 OpenFlow 进行交互.由于每层控制节点维持的网络视图是抽象的网络拓扑,因此,在执行跨域网络应用,如域间路由优化时,其计算对象也是抽象拓扑,此时,计算结果在抽象的拓扑中是最优的,而非全局最优.

Kandoo^[58]是分层控制平面的先驱,它将控制平面分为两层,其中,控制器分为根控制器和本地控制器两类.本地控制器仅维护局部网络视图,以完全分布式的形式组织,主要处理数量较多的小流,因为这些流在网络中的流量较小,非最优的转发路径不会带来很大的影响;根控制器则维护全局网络视图,负责数量较少但流量较大的大象流,对其进行全局最优处理,防止网络传输性能恶化.

xBar^[59]提出在 SDN 网络中采用分层、聚合、递归等概念,将分层控制的思想进一步完善.Orion^[60,61]设计了详细的分层控制平面系统架构和功能模块,并给出一种旨在解决分层控制系统中跨域路径非最优问题的跨域通信算法.该算法在父节点控制器抽象其控制域时包含了更多的内部拓扑信息,使其能够利用这些额外信息优化路径计算结果.这也是优化分层控制平面网络处理算法的常用思路.

分层控制平面由于采用递归控制机制,要求总的控制层数不能过多,否则会造成较大的处理延迟.但是,控制层数减少会导致每个控制器控制的设备过多、负载过重,同样会导致处理延迟.文献[62]对控制平面的层数和控制域大小进行折中分析,为分层控制平面结构的优化设计提供了参考.

与完全分布式的控制平面相比,分层控制平面保留了集中控制的优势,其网络处理收敛速度较快,对网络产生的信令负担小.与逻辑集中、物理分布的控制平面相比,分层控制平面由于树形组织方式和底层控制器完全分布式的特点,使得网络视图更新不再需要控制器间过多的信息交互;对下属网络域的抽象使得计算转发策略时不再需要考虑全网节点,极大地提高了算法的可扩展性.另外,逐级递归控制也使得网络事件广播保持在特定的网络域内,不会扩展至其他域;最后,对不同层级网络视图的抽象,在简化网络管理的同时也隐藏了网络内部细节,保障了自主性和安全性.因此,分层控制平面具有良好的可扩展性,是未来大规模 SDN 网络的发展方向.

5 控制资源优化分配

分布式控制平面虽然能够有效地解决可扩展性问题,但是这种简单的、通过提高控制器数量和性能来保障可扩展性的方法缺乏必要的管理和调节机制,需要根据底层网络视图和流量分布对控制资源进行优化分配,提高控制平面的资源利用率,使单位控制资源投入带来的可扩展性改善程度提高.当前,控制资源优化分配的研究主要包括控制器优化放置和弹性控制两类.

5.1 控制器优化放置

由于网络中不同位置的网络事件均需要上传至控制器进行处理,控制器与所属交换机的距离会影响网络事件的传输时延,因此,控制器的放置对网络请求的响应时间有比较明显的影响.对于分布式控制平面,需要根据数据平面拓扑与控制负载分布来确定所需控制器的数量、放置及控制器与交换机的映射关系,以尽可能小的成本实现所需的处理性能.此外,控制器的优化放置除了对控制平面的可扩展性产生影响外,还会影响 SDN 网络的可靠性^[63]等性能.本文主要针对可扩展性相关的控制器放置问题展开探讨,该类问题中,控制器-交换机之间距离、负载均衡性能和控制资源成本是常用的衡量因素.

文献[64]首次描述了 SDN 网络中控制器的放置问题,根据控制器之间或控制器与交换机之间的传输时延判断网络中需要的控制器个数及控制器在拓扑中的位置.该文指出:对于众多中等规模的网络,一个控制器即可满足传输时延的要求;当网络规模进一步扩大后,由于控制器与交换机之间的传输时延约束,必须在网络中部署

多个控制器,该文给出了对控制器放置问题进行进一步研究的必要性。

文献[65]在文献[64]的基础上,以最小化最大的控制器-交换机传输时延作为优化目标,同时考虑控制器的容量约束,对分布式控制平面的控制器放置问题展开研究,该问题被建模为有容量约束的 k -center 问题。文献[66]从负载均衡的角度出发,通过构建控制树进行控制器的放置,但在该文中,控制负载的衡量指标是交换机的数目。实际上,由于所连接主机的功能不同,不同交换机造成的控制负载千差万别,以交换机产生的网络请求数目来衡量控制负载更为合理。文献[67]就是以此衡量控制负载,并根据 Little 公式计算控制器负载造成的处理时延,与控制器-交换机距离带来的传输时延共同组成网络请求响应时延,作为分布式控制平面控制器放置问题的限制条件。如此设计同时满足了控制器-交换机距离和负载均衡约束,使得控制资源的分配与数据平面的处理需求分布匹配,以较小的控制资源成本实现了 SDN 网络的需求,提高了控制平面的资源利用率。文献[68]进一步考虑了分布式控制平面中控制器之间的响应性能,使用讨价还价博弈对控制器-交换机传输时延和控制器之间的传输时延进行权衡分析,设计了一种控制器放置方案。文献[69]在综合考虑上述因素的情况下,设计了一种基于交换机分布密度的聚类算法。首先将数据平面分域,再在各个域中确定其最优控制器。该算法计算复杂度较低,能够适用于大规模网络。文献[70]则探讨了一种增量部署场景,即在现有 SDN 数据平面和控制器部署的基础上,当数据平面规模扩大时,如何以最小的成本增量式地部署足够多的新控制器,这给 SDN 网络的管理提供了更多的思路。

5.2 弹性控制

网络流量具有空间分布不均和时域动态变化的特征^[30],体现到 SDN 网络,就是空间上不同交换机由于所连接主机的需求和提供的服务不同,流到达速率不同;时域上不同交换机的流到达速率按照网络需求的动态变化而变化。由于用户行为的随机性,网络流具有随机性和瞬时突发特征^[30,45]。分布式控制平面中固定配置的控制平面很难适应这一变化,往往导致部分控制器发生负载过重的情况,使网络事件无法得到及时响应,甚至被拒绝响应。

应对负载增加的通用思路是增加相应的处理资源,即:在控制器负载过重时对控制区域进行裂变,并增加新的控制器以适应负载的变化。这种方式虽然能够解决控制器负载过重的问题,但是对问题的考虑是一种局部的不经济的思路。应该注意到:整个 SDN 网络中,控制负载具有空间分布不均的特征,使得控制平面负载不均衡,某些控制器负载过重的同时,其他控制器可能处于空闲状态,现有控制平面的资源并未得到充分的利用。这是由于控制器-交换机映射与负载空间分布不匹配造成的。并且,由于负载分布随时间随机变化,并不能通过静态的最优映射^[65]达到一劳永逸的目的。因此,更为经济优化的思路是根据负载的分布实时动态地调整控制器与交换机的映射,实现控制平面负载均衡。这种思路也称为弹性控制^[71]。

SDN 中,控制平面将控制资源分配到每个交换机,以处理数据平面产生的控制负载。弹性控制的思路是:根据控制负载的分布,动态地、自适应地调整控制资源在数据平面的分布。这一分布包括控制器与交换机的映射关系和控制器与所属交换机的相对位置。前者代表控制资源在数据平面的数量分布,即,交换机所拥有的控制资源总量;后者代表控制资源在数据平面的位置分布,即,交换机与其所属控制资源的距离。相应地,弹性控制技术包含对控制资源数量分布和位置分布的调整,这一调整可以通过动态部署控制平面和动态更改控制器与交换机的映射关系来实现。

DCPP^[72]构建了一种控制平面动态部署方案,它根据数据平面的负载分布,动态调整控制器的数目、位置以及与交换机的映射,使其与负载的空间分布完全匹配。LiDy+^[73,74]通过添加对控制负载的预测来预留足够的控制资源,根据控制负载的分布,动态调整控制器的部署,以期在控制平面成本和控制器交换机的时延之间取得折中。控制平面动态部署方案是最优的弹性控制方案,但是,这种方案需要频繁地重新部署控制平面,会带来大量的管理开销和时延,无法实时响应控制负载分布的变化,影响到网络的正常运行。

ElastiCon^[71,75]给出一种温和的解决方案,它不改变当前控制平面的规模和部署,仅仅通过使交换机在控制域之间迁移来调整控制器的负载,实现负载均衡。当且仅当数据平面的整体负载与控制平面的整体处理能力不匹配时,才会调整控制器的数目。ElastiCon 还给出了无缝的交换机迁移协议及包括交换机迁移、控制器池变更

在内的弹性控制算法。

基于 ElastiCon 给出的交换机迁移协议,我们希望设计控制器-交换机重映射算法,计算最优的控制器-交换机的映射,使其与负载分布相匹配。文献[76]以控制平面负载均衡为优化目标对这一问题进行建模,并使用线性规划给出其近似解,但是线性规划求解复杂度较高,且随着网络规模的扩大,算法的可扩展性较差。文献[77]以控制资源利用率最大化为优化目标展开求解,将动态重映射过程建模为马尔可夫链,并设计了一种分布式跳转算法,将计算任务分布在不同的控制器。该算法充分发挥了各个控制器的计算能力,但是整个重映射过程信令比较复杂,收敛速度较慢。文献[78]则以最小化控制平面响应时间为目标,将重映射过程建模为多对一稳定匹配问题进行求解,之后使用组合博弈实现负载均衡。由于第 1 步缩减了组合博弈的搜索空间,使得整个求解过程迅速收敛,保障了算法的实时性,提高了算法应对数据流分布随机变化的适用性和应用于大规模网络的可扩展性。

6 总结与展望

6.1 总结

SDN 控制转发分离的特性简化了网络的配置和管理,同时也带来了严重的可扩展性问题。本文结合分布式系统可扩展性的衡量指标来分析 SDN 控制平面的可扩展性,并给出了 4 种改善 SDN 控制平面可扩展性的思路。在此基础上,本文从数据平面缓存优化、高性能控制器、分布式控制平面和控制资源优化分配 4 种技术路线出发,分析探讨了现有 SDN 控制平面可扩展性问题的主要解决方案和研究进展。这 4 种技术路线与第 1.3 节中 4 种改善思路的对应关系见表 1。

Table 1 Correspondence between the 4 technical routes and the 4 improvement strategies

表 1 4 种技术路线与 4 种改善思路的对应关系

技术路线	改善思路
数据平面缓存优化	(1) 优化管理交换机中缓存的流表项
高性能控制器	(2) 提高控制平面的处理能力
分布式控制平面	(2) 提高控制平面的处理能力;(3) 减小控制平面和交换机之间的传输时延
控制资源优化分配	(3) 减小控制平面和交换机之间的传输时延;(4) 提高现有控制资源的利用率

6.2 展望

当前,随着 SDN 网络的逐步推广,SDN 控制平面面临的可扩展性问题越来越严峻。尽管当前针对这一问题已经提出了一系列的解决方案,但是仍有许多关键问题亟待解决。此外,目前,SDN 还不是一个成熟的架构,随着 SDN 的持续演进,新的控制平面可扩展性解决方案会相继涌现,同时,对控制平面可扩展性问题的研究也会进一步推动 SDN 架构的不断创新。下面对未来研究重点和发展趋势进行探讨。

(1) 基于网络需求确定网络事件的处理粒度

灵活的网络控制是 SDN 的一个重要特征。这一特征主要是通过细粒度的网络请求和被动的规则安装实现的,而这两个因素对控制平面带来了较重的控制负载,是控制平面可扩展性问题的源头。然而,实际网络中,用户的通信需求往往是基于应用或服务进行划分的,网络核心对数据流的处理往往也基于聚合的原则。因此,完全细粒度的处理并不是必要的,基于网络需求来确定网络事件的处理粒度才是更合理的选择,即,对网络事件进行合理的抽象。当前,对 SDN 抽象方案的研究主要集中于高级编程语言的设计,将抽象思路拓展到南向接口,对网络控制的灵活性、数据平面和控制平面的可扩展性均具有现实的意义。

(2) 标准化交换机的控制功能

SDN 控制转发分离的思路简化了网络的配置和管理,但是很多交换机本地即可完成的控制功能只能上传至控制平面处理,给控制平面带来了较大的负担,也延长了这些网络功能的响应时间。当前,很多研究者对在交换机实现部分控制功能展开了研究,但是,由于这些方案对 OpenFlow 交换机结构和应用模式进行了修改,其推广难度较大。因此,有必要对交换机本地需要且在本地即可完成的控制功能集合进行提取,并以标准化的形式整合到 OpenFlow 交换机的软件协议栈中。应该注意的是:这一设计并不违反控制转发分离的原则,网络的控制转

发分离仍然以软硬件分离的形式实现,只是控制平面结构得到了革新,以分层控制平面的形式呈现。

(3) 弹性控制平面的资源池模型

ElastiCon^[71]引用云计算中资源池化的概念对控制器池进行了描述.实际上,由于数据中心网络中虚拟化技术的发展,对 SDN 控制器的管理也可以采用资源池化的方式展开,在空间尺度和性能尺度上扩展控制平面的处理能力.同时,为了灵活、高效地管理控制资源,可以基于服务化的思想设计分布式服务框架,对控制平面功能进行模块化抽象和组合.在这一框架下,对高效弹性控制平面、控制器部署和迁移、控制功能模块化抽象、控制功能组合等问题展开研究具有重要的意义。

(4) 面向分布式控制平面的分布式算法设计

传统网络中,由于网络设备完全分布式的部署,网络协议往往通过分布式算法实现.SDN 网络出于可扩展性考虑,引入了分布式的控制平面.对于完全分布式的控制平面和分层控制平面,在处理跨域网络事件时,由于不具备全局网络视图,往往需要沿用传统网络的分布式算法进行处理.然而,SDN 控制器仍具有域内集中控制的优势,如何利用这一优势对 SDN 分布式控制平面设计优化的分布式算法、提高控制平面可扩展性,也是未来的一个研究方向。

(5) 数据平面的多维划分

SDN 网络中,为了提高控制平面的可扩展性,往往将数据平面负载划分给多个控制器分别处理.现有研究中,主要以交换机作为数据平面划分的粒度.实际上,除了根据物理拓扑进行划分,数据平面负载还可以有多种划分方式,如根据虚拟网进行的虚拟拓扑划分、根据网络应用不同进行的流表空间划分等.不同的划分方式和划分粒度适用于不同的网络需求,有必要根据网络需求设计合理的数据平面划分,提高控制平面的处理性能和可扩展性。

(6) 网络应用感知的控制资源优化分配

SDN 控制平面的可扩展性关注的主体是网络请求,现有研究在评估控制平面负载时,也往往以网络请求的数量为依据.然而,对不同的网络请求,控制平面需要调用不同的网络应用进行处理,消耗的控制资源也就有所不同.部分应用,如网络虚拟化和流量工程等,其计算复杂度较高,消耗的控制资源较多;而接入控制、MAC 学习等应用消耗的资源较少.因此,在进行控制资源分配时,应综合考虑网络请求的资源消耗及其在网络中的分布特征,确定优化的控制资源分配方案,提高控制资源的利用率。

References:

- [1] Internet World Stats. Internet growth statistics. <http://www.internetworldstats.com/emarketing.htm>
- [2] Cisco. Cisco visual networking index: Forecast and methodology, 2013~2018. Technical Report, 2013.
- [3] Chowdhury NM, Boutaba R. Network virtualization: State of the art and research challenges. *IEEE Communications Magazine*, 2009,47(7):20-26. [doi: 10.1109/MCOM.2009.5183468]
- [4] Chowdhury NM, Kabir M, Boutaba R. A survey of network virtualization. *Computer Networks*, 2010,54(5):862-876. [doi: 10.1016/j.comnet.2009.10.017]
- [5] McKeown N, Anderson T, Balakrishnan H, Guru P, Larry P, Jennifer R, Scott S, Jonathan T. OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 2008,38(2):69-74. [doi: 10.1145/1355734.1355746]
- [6] McKeown N. Software-Defined networking. *INFOCOM Keynote Talk*, 2009,17(2):30-32.
- [7] Zhang CK, Cui Y, Tang HY, Wu JP. State-of-the-Art survey on software-defined networking (SDN). *Ruan Jian Xue Bao/Journal of Software*, 2015,26(1):62-81 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4701.html> [doi: 10.13328/j.cnki.jos.004701]
- [8] Kreutz D, Ramos FMV, Verissimo PE, Rothenberg CE, Azodolmolky S, Uhlig S. Software-Defined networking: A comprehensive survey. *Proc. of the IEEE*, 2015,103(1):14-76. [doi: 10.1109/JPROC.2014.2371999]
- [9] Nunes BAA, Mendonca M, Nguyen X, Obraczka K, Turlitti T. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 2014,16(3):1617-1634. [doi: 10.1109/SURV.2014.012214.00180]

- [10] Zuo QY, Chen M, Zhao GS, Xing CY, Zhang GM, Jiang PC. Research on openflow-based sdn technologies. Ruan Jian Xue Bao/ Journal of Software, 2013,24(5):1078–1097 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4390.html> [doi: 10.3724/SP.J.1001.2013.04390]
- [11] Xie J, Guo D, Hu Z, Qu T, Lv P. Control plane of software defined networks: A survey. Computer Communications, 2015,67:1–10. [doi: 10.1016/j.comcom.2015.06.004]
- [12] OpenFlow. OpenFlow switch specification. <https://www.opennetworking.org/sdn-resources/technical-library>
- [13] Jain S, Kumar A, Mandal S, Ong J, Poutievski L, Singh A, Venkata S, Wanderer J, Zhou J, Zhu M, Zolla J, Hözl U, Stuart S, Vahdat A. B4: Experience with a globally-deployed software defined WAN. ACM SIGCOMM Computer Communication Review, 2013,43(4):3–14. [doi: 10.1145/2534169.2486019]
- [14] Jain S, Kumar A, Mandal S, Ong J, Poutievski L, Singh A, Venkata S, Wanderer J, Zhou JL, Zhu M, Zolla J, Hözl U, Stuart S, Vahdat A. The nature of data center traffic: Measurements & analysis. In: Proc. of the 2009 ACM SIGCOMM Conf. on Internet Measurement Conf. (IMC). Chicago: ACM Press, 2009. 202–208. [doi: 10.1145/1644893.1644918]
- [15] Tootoonchian A, Gorbunov S, Ganjali Y, Casado M, Sherwood R. On controller performance in software-defined networks. In: Proc. of the 2nd USENIX Conf. on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (HotICE). San Jose: USENIX, 2012. 7–10. <https://www.usenix.org/conference/hot-ice12/workshop-program/presentation/tootoonchian>
- [16] Tootoonchian A, Gorbunov S, Ganjali Y, Casado M, Sherwood R. Applying NOX to the datacenter. In: Proc. of the 2009 ACM Workshop on Hot Topics in Networks Workshop (HotNets). New York: ACM Press, 2009. 1–6. <http://conferences.sigcomm.org/hotnets/2009/program.html>
- [17] Berde P, Gerola M, Hart J, Higuchi Y, Kobayashi M, Koide T, Lantz B, O'Connor B, Radoslavov P, Snow W, Parulkar G. ONOS: Towards an open, distributed SDN OS. In: Proc. of the 3rd Workshop on Hot Topics in Software Defined Networking (HotSDN). Chicago: ACM Press, 2014. 1–6. [doi: 10.1145/2620728.2620744]
- [18] Yeganeh SH, Tootoonchian A, Ganjali Y. On scalability of software-defined networking. IEEE Communications Magazine, 2013, 51(2):136–141. [doi: 10.1109/MCOM.2013.6461198]
- [19] Neuman BC. Scale in distributed systems. Readings in Distributed Computing Systems, 1995,74(2):463–489.
- [20] Xu K, Zhu M, Lin C. Internet architecture evaluation models, mechanism, and methods. Ji Suan Ji Xue Bao/Chinese Journal of Computers, 2012,35(10):1985–2006 (in Chinese with English abstract). <http://cjic.ict.ac.cn/qwjs/view.asp?id=3647> [doi: 10.3724/SP.J.1016.2012.01985]
- [21] Hu J, Lin C, Li X, Huang J. Scalability of control planes for software defined networks: Modeling and evaluation. In: Proc. of the IEEE 22nd Int'l Symp. of Quality of Service (IWQoS). Hong Kong: IEEE, 2014. 147–152. [doi: 10.1109/IWQoS.2014.6914314]
- [22] Katta N, Rexford J, Walker D. Infinite cache-flow in software-defined networks. In: Proc. of the 2014 ACM Workshop on Hot Topics in Software Defined Networking (HotSDN). Chicago: ACM Press, 2014. 175–180. [doi: 10.1145/2620728.2620734]
- [23] Wolfgang B, Menth M, Felzer W, Dixon C, Carter J. Wildcard compression of inter-domain routing tables for OpenFlow based software-defined networking. In: Proc. of the 3rd EUR. Workshop Software-Defined Networking (EWSDN). Budapest: IEEE, 2014. 25–30. [doi: 10.1109/EWSDN.2014.23]
- [24] Rifai M, Huin N, Caillouet C, Giroire F, Pacheco DML, Moulrierac J, Guillaume UK. Too many SDN rules? Compress them with MINNIE. In: Proc. of the 2015 IEEE Global Communications Conf. (GLOBECOM). San Diego: IEEE, 2015. 1–7. [doi: 10.1109/GLOCOM.2014.7417661]
- [25] Meiners CR, Liu AX, Torng E. Bit weaving: A non-prefix approach to compressing packet classifiers in TCAMs. IEEE/ACM Trans. on Networking, 2009,20(2):488–500. [doi: 10.1109/TNET.2011.2165323]
- [26] Leng B, Huang L, Qiao C, Xu H. A decision-tree-based on-line flow table compressing method in software defined networks. In: Proc. of the 2016 IEEE/ACM Int'l Symp. on Quality of Service (IWQoS). Vilanova: ACM Press, 2016. 1–2. [doi: 10.1109/IWQoS.2016.7590401]
- [27] Giroire F, Havet F, Moulrierac J. Compressing two-dimensional routing tables with order. Electronic Notes in Discrete Mathematics, 2015,52(47):351–358. [doi: 10.1016/j.endm.2016.03.046]
- [28] Huang M, Liang W, Xu Z, Xu W, Guo S, Xu Y. Dynamic routing for network throughput maximization in software-defined networks. In: Proc. of the 2016 IEEE Conf. on Computer Communications (INFOCOM). San Francisco: IEEE, 2016. 1–9. [doi: 10.1109/INFOCOM.2016.7524613]

- [29] Kannan K, Banerjee S. FlowMaster: Early eviction of dead flow on SDN switches. In: Proc. of the 15th Int'l Conf. on Distributed Computing and Networking (ICDCN). Coimbatore: Springer-Verlag, 2014. 484–498. [doi: 10.1007/978-3-642-45249-9_32]
- [30] Benson T, Akella A, Maltz DA. Network traffic characteristics of data centers in the wild. In: Proc. of the 10th ACM SIGCOMM Conf. on Internet Measurement (IMC). Melbourne: ACM Press, 2010. 267–280. [doi: 10.1145/1879141.1879175]
- [31] Zhang L, Lin R, Xu S, Wang S. AHTM: Achieving efficient flow table utilization in software defined networks. In: Proc. of the 2014 IEEE Global Communications Conf. (GLOBECOM). Austin: IEEE, 2014. 1897–1902. [doi: 10.1109/GLOCOM.2014.7037085]
- [32] Zhang L, Wang S, Xu S, Lin R. TimeoutX: An adaptive flow table management method in software defined networks. In: Proc. of the 2015 IEEE Global Communications Conf. (GLOBECOM). San Diego: IEEE, 2015. 1–6. [doi: 10.1109/GLOCOM.2015.7417563]
- [33] Liang H, Hong P, Li J, Ni D. Effective idle_timeout value for instant messaging in software defined networks. In: Proc. of the 2015 IEEE Int'l Conf. on Communication Workshop (ICC WKSHP). London: IEEE, 2015. 352–356. [doi: 10.1109/ICCW.2015.7247204]
- [34] Huang H, Guo S, Li P, Liang W, Zomaya AY. Cost minimization for rule caching in software defined networking. IEEE Trans. on Parallel and Distributed Systems, 2016,27(4):1007–1016. [doi: 10.1109/TPDS.2015.2431684]
- [35] Yu M, Rexford J, Freedman MJ, Wang J. Scalable flow-based networking with DIFANE. ACM SIGCOMM Computer Communication Review, 2010,40(4):351–362. [doi: 10.1145/1851182.1851224]
- [36] Curtis AR, Mogul JC, Tourrilhes J, Yalagandula P, Sharma P, Banerjee S. DevoFlow: Scaling flow management for high-performance networks. ACM SIGCOMM Computer Communication Review, 2011,41(4):254–265. [doi: 10.1145/2018436.2018466]
- [37] Moshref M, Bhargava A, Gupta A, Yu M, Govindan R. Flow-Level state transition as a new switch primitive for SDN. In: Proc. of the 2014 ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN). Chicago: ACM Press, 2014. 61–66. [doi: 10.1145/2620728.2620729]
- [38] Zhu S, Bi J, Sun C, Wu C, Hu H. SDPA: Enhancing stateful forwarding for software-defined networking. In: Proc. of the 23rd IEEE Int'l Conf. on Network Protocols (ICNP). San Francisco: IEEE, 2015. 10–13. [doi: 10.1109/ICNP.2015.45]
- [39] Bi J, Zhu S, Sun C, Yao G, Hu H. Supporting virtualized network functions with stateful data plane abstraction. IEEE Network, 2016,30(3):40–45. [doi: 10.1109/MNET.2016.7474342]
- [40] Newport C, Zhou W. The (surprising) computational power of the SDN data plane. In: Proc. of the 2015 IEEE Conf. on Computer Communications (INFOCOM). Hong Kong: IEEE, 2015. 496–504. [doi: 10.1109/INFOCOM.2015.7218416]
- [41] NICIRA NETWORKS. Open vSwitch, an open virtual switch. <http://openvswitch.org/>
- [42] Cai Z, Cox AL, Ng TSE. Maestro: A system for scalable OpenFlow control. TSEN Maestro Technical Report, TR10-08, Houston: Rice University, 2010.
- [43] Erickson D. The Beacon OpenFlow controller. In: Proc. of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN). Hong Kong: ACM Press, 2013. 13–18. [doi: 10.1145/2491185.2491189]
- [44] Voellmy A, Wang J. Scalable software defined network controllers. ACM SIGCOMM Computer Communication Review, 2012, 42(4):289–290. [doi: 10.1145/2377677.2377735]
- [45] Lin CT, Wu CM, Huang M, Wen ZY, Zheng QH. Performance evaluation for SDN deployment: An approach based on stochastic network calculus. China Communications, 2016,13(s1):98–106. [doi: 10.1109/CC.0.7560881]
- [46] Koponen T, Casado M, Gude N, Stribling J, Poutievski L, Zhu M, Ramanathan R, Iwata Y, Inoue H, Hama T, Shenker S. Onix: A distributed control platform for large-scale production networks. In: Proc. of the 9th USENIX Symp. on Operating Systems Design and Implementation (OSDI). Vancouver: USENIX, 2010. 351–364. <https://dl.acm.org/citation.cfm?id=1924968>
- [47] Tootoonchian A, Ganjali Y. HyperFlow: A distributed control plane for OpenFlow. In: Proc. of the 2010 Internet Network Management Workshop on Research on Enterprise Networking (INM/WREN). San Jose: USENIX, 2010. 3. <https://www.usenix.org/legacy/events/inmwren10/tech/tech.html#Tootoonchian>
- [48] Phemius K, Bouet M, Leguay J. DISCO: Distributed multi-domain SDN controllers. In: Proc. of the 2014 IEEE/IFIP Network Operations and Management Symp. (NOMS). Krakow: IEEE, 2014. 1–4. [doi: 10.1109/NOMS.2014.6838330]
- [49] OpenContrail. <http://www.opencontrail.org/>

- [50] Panda A, Zheng W, Hu X, Krishnamurthy A, Shenker S. SCL: Simplifying distributed SDN control planes. In: Proc. of the 14th USENIX Symp. on Networked Systems Design and Implementation (NSDI). Boston: USENIX, 2017. 1–17. <https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/panda-aurojit-scl>
- [51] Canini M, Kuznetsov P, Levin D, Schmid S. A distributed and robust SDN control plane for transactional network updates. In: Proc. of the 2015 IEEE Conf. on Computer Communications (INFOCOM). Hong Kong: IEEE, 2015. 190–198. [doi: 10.1109/INFOCOM.2015.7218382]
- [52] Li JF, Lan JL, Hu YX, Wu JX. Quantitative approach of multi-controller's consensus in SDN. *Tong Xin Xue Bao/Journal on Communications*, 2016,37(6):86–93 (in Chinese with English abstract). <http://www.infocomm-journal.com/txxb/CN/10.11959/j.issn.1000-436x.2016119> [doi: 10.11959/j.issn.1000-436x.2016119]
- [53] Levin D, Wundsam A, Heller B, Handigol N, Feldmann A. Logically centralized? State distribution trade-offs in software defined networks. In: Proc. of the 2012 ACM on Hot Topics in Software Defined Networking (HotSDN). Helsinki: ACM Press, 2012. 1–6. [doi: 10.1145/2342441.2342443]
- [54] Tam AW, Xi K, Chao HJ. Use of devolved controllers in data center networks. In: Proc. of the IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS). Shanghai: IEEE, 2011. 596–601. [doi: 10.1109/INFOCOMW.2011.5928883]
- [55] Schmid S, Suomela J. Exploiting locality in distributed SDN control. In: Proc. of the 2013 ACM on Hot Topics in Software Defined Networking (HotSDN). Hong Kong: ACM Press, 2013. 121–126. [doi: 10.1145/2491185.2491198]
- [56] Lin P, Bi J, Chen Z, Wang Y, Hu H, Xu A. WE-Bridge: West-East bridge for SDN inter-domain network peering. In: Proc. of the 2014 IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS). Toronto, 2014. 111–112. [doi: 10.1109/INFOCOMW.2014.6849180]
- [57] Yin H, Xie H, Tsou T, Lopez D, Aranda P, Sidi R. SDNi: A message exchange protocol for software defined networks (SDNs) across multiple domains. IETF draft, 2012.
- [58] Yeganeh SH, Ganjali Y. Kandoo: A framework for efficient and scalable offloading of control applications. In: Proc. of the 2012 ACM on Hot Topics in Software Defined Networking (HotSDN). Helsinki: ACM Press, 2012. 19–24. [doi: 10.1145/2342441.2342446]
- [59] McCauley J, Panda A, Casado M, Koponen T, Shenker S. Extending SDN to large-scale networks. In: Proc. of the Open Networking Summit 2013. Santa Clara, 2013. 1–2. <https://events.linuxfoundation.org/events/open-networking-summit-europe-2018/>
- [60] Fu Y, Bi J, Chen Z, Gao K, Zhang B, Chen G, Wu J. A hybrid hierarchical control plane for flow-based large-scale software-defined networks. *IEEE Trans. on Network and Service Management*, 2015,12(2):117–131. [doi: 10.1109/TNSM.2015.2434612]
- [61] Fu Y, Bi J, Gao K, Chen Z, Wu J, Hao B. Orion: A hybrid hierarchical control plane of software-defined networking for large-scale networks. In: Proc. of the 22nd IEEE Int'l Conf. on Network Protocols (ICNP). Research Triangle Park: IEEE, 2014. 569–576. [doi: 10.1109/ICNP.2014.91]
- [62] Liu Y, Hecker A, Guerzoni R, Despotovic Z, Beker S. On optimal hierarchical SDN. In: Proc. of the 2015 IEEE Int'l Conf. on the Communications (ICC). London: IEEE, 2015. 5374–5379. [doi: 10.1109/ICC.2015.7249178]
- [63] Hu Y, Wang W, Gong X, Que X, Cheng S. On reliability-optimized controller placement for software-defined networks. *China Communications*, 2014,11(2):38–54. [doi: 10.1109/CC.2014.6821736]
- [64] Heller B, Sherwood R, McKeown N. The controller placement problem. In: Proc. of the 2012 ACM Hot Topics on Software Defined Networking (HotSDN). Helsinki: ACM Press, 2012. 7–12. [doi: 10.1145/2342441.2342444]
- [65] Yao G, Bi J, Li Y, Guo L. On the capacitated controller placement problem in software defined networks. *IEEE Communications Letters*, 2014,18(8):1339–1342. [doi: 10.1109/LCOMM.2014.2332341]
- [66] Jimenez Y, Cervello-Pastor C, Garcia AJ. On the controller placement for designing a distributed SDN control layer. In: Proc. of the 2014 IFIP Networking Conf. Stockholm: IEEE, 2014. 1–9. [doi: 10.1109/IFIPNetworking.2014.6857117]
- [67] Cheng TY, Wang M, Jia X. QoS-Guaranteed controller placement in SDN. In: Proc. of the 2015 IEEE Global Communications Conf. (GLOBECOM). San Diego: IEEE, 2015. 1–6. [doi: 10.1109/GLOCOM.2015.7416960]
- [68] Ksentini A, Bagaa M, Taleb T, Balasingham I. On using bargaining game for optimal placement of SDN controllers. In: Proc. of the 2016 IEEE Int'l Conf. on Communications (ICC). Kuala Lumpur: IEEE, 2016. 1–6. [doi: 10.1109/ICC.2016.7511136]

- [69] Liao JX, Sun HF, Wang JY, Qi Q, Li K, Li TH. Density cluster based approach for controller placement problem in large-scale software defined networkings. *Computer Networks*, 2017,112(15):24–35. [doi: 10.1016/j.comnet.2016.10.014]
- [70] Sallahi A, St-Hilaire M. Expansion model for the controller placement problem in software defined networks. *IEEE Communications Letters*, 2017,21(2):274–277. [doi: 10.1109/LCOMM.2016.2621746]
- [71] Dixit A, Hao F, Mukherjee S, Lakshman T, Kompella RR. Towards an elastic distributed SDN controller. *ACM SIGCOMM Computer Communication Review*, 2013,43(4):7–12. [doi: 10.1145/2534169.2491193]
- [72] Bari MF, Roy AR, Chowdhury SR, Zhang Q, Zhani MF, Ahmed R, Boutaba R. Dynamic controller provisioning in software defined networks. In: *Proc. of the 9th Int'l Conf. on Network and Service Management (CNSM)*. Zurich: IEEE, 2013. 18–25. [doi: 10.1109/CNSM.2013.6727805]
- [73] Ul Huque MTI, Jourjon G, Gramoli V. Revisiting the controller placement problem. In: *Proc. of the IEEE 40th Conf. on Local Computer Networks (LCN)*. Clearwater Beach: IEEE, 2015. 450–453. [doi: 10.1109/LCN.2015.7366350]
- [74] Huque MTIU, Si W, Jourjon G, Gramoli V. Large-Scale dynamic controller placement. *IEEE Trans. on Network and Service Management*, 2017,14(1):63–76. [doi: 10.1109/TNSM.2017.2651107]
- [75] Dixit AA, Hao F, Mukherjee S, Lakshman T, Kompella RR. ElastiCon: An elastic distributed SDN controller. In: *Proc. of the 10th ACM/IEEE Symp. on Architectures for Networking and Communications Systems (ANCS)*. Los Angeles: ACM Press, 2014. 17–28. [doi: 10.1145/2658260.2658261]
- [76] Liang W, Gao X, Wu F, Chen G, Wei W. Balancing traffic load for devolved controllers in data center networks. In: *Proc. of the 2014 IEEE Global Communications Conf. (GLOBECOM)*. Austin: IEEE, 2014. 2258–2263. [doi: 10.1109/GLOCOM.2014.7037144]
- [77] Cheng G, Chen H, Wang Z, Chen S. DHA: Distributed decisions on the switch migration toward a scalable SDN control plane. In: *Proc. of the 2015 IFIP Networking Conf.* Toulouse: IEEE, 2015. 1–9. [doi: 10.1109/IFIPNetworking.2015.7145319]
- [78] Wang T, Liu FM, Guo J, Xu H. Dynamic SDN controller assignment in data center networks: Stable matching with transfers. In: *Proc. of the 2016 IEEE Conf. on Computer Communications (INFOCOM)*. San Francisco: IEEE, 2016. 1–9. [doi: 10.1109/INFOCOM.2016.7524357]

附中文参考文献:

- [7] 张朝昆,崔勇,唐嵩嵩,吴建平.软件定义网络(SDN)研究进展.软件学报,2015,26(1):62–81. <http://www.jos.org.cn/1000-9825/4701.html> [doi: 10.13328/j.cnki.jos.004701]
- [10] 左青云,陈鸣,赵广松,邢长友,张国敏,蒋培成.基于 OpenFlow 的 SDN 技术研究.软件学报,2013,24(5):1078–1097. <http://www.jos.org.cn/1000-9825/4390.html> [doi: 10.3724/SP.J.1001.2013.04390]
- [20] 徐格,朱敏,林闯.互联网体系结构评估模型、机制及方法研究综述.计算机学报,2012,35(10):1985–2006. <http://cjc.ict.ac.cn/qwjs/view.asp?id=3647> [doi: 10.3724/SP.J.1016.2012.01985]
- [52] 李军飞,兰巨龙,胡宇翔,邬江兴.SDN 多控制器一致性的量化研究.通信学报,2016,37(6):86–93. <http://www.infocomm-journal.com/txxb/CN/10.11959/j.issn.1000-436x.2016119> [doi: 10.11959/j.issn.1000-436x.2016119]



张少军(1989—),男,山西运城人,博士生,主要研究领域为新型网络体系结构,软件定义网络.



胡宇翔(1982—),男,博士,副研究员,主要研究领域为新一代信息网络关键理论与技术.



兰巨龙(1962—),男,博士,教授,博士生导师,主要研究领域为宽带信息网络,新一代信息网络关键理论与技术.



江逸茗(1984—),男,博士,助理研究员,主要研究领域为新型网络体系结构,网络虚拟化.